

Ahmat Barkai Moussa

Task 11 Süre : 1 Gün - ProductModel üzerinde yeni bir alan açın ve bu alanın solr'a indexlenmesini sağlayın. Indexlediğiniz bu alanın ürün liste sayfasında görünmesini sağlayın.

```
<itemtype code="Product" autocreate="false" generate="false">
  <attributes>
    <attribute qualifier="productOriginName" type="java.lang.String">
      <description>Origin Country Name </description>
      <persistence type="property"/>
      <modifiers read="true" write="true" optional="true" search="true"/>
    </attribute>

    <attribute qualifier="isDisasterAidProduct" type="java.lang.Boolean">
      <defaultvalue>Boolean.FALSE</defaultvalue>
      <persistence type="property"/>
      <modifiers optional="true"/>
    </attribute>
  </attributes>
</itemtype>
```

Product itemType'ımıza attribute'umuzu ekliyoruz .

```
<!-->
<beans xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="beans.xsd">
  <bean class="de.hybris.platform.commercefacades.product.data.ProductData">
    <property name="productOriginName" type="java.lang.String"/>
  </bean>
</beans>
```

attributu avansasfacadebeans productData beans'ine ekliyoruz

```
@Override
public void populate(final ProductModel source, final ProductData target) {
    super.populate(source, target);

    target.setProductOriginName(source.getProductOriginName());
}
```

Attributu populate ediyoruz AvsProductPopulator'da

Sonra Provider oluşturuyoruz :

```
1 usage
public class ProductOriginNameValueProvider extends AbstractPropertyFieldValueProvider implements FieldValueProvider, Serializable {

    3 usages
    private FieldNameProvider fieldNameProvider;

    @Override
    public Collection<FieldValue> getFieldValues(final IndexConfig indexConfig, final IndexedProperty indexedProperty,
                                                final Object model) throws FieldValueProviderException {

        final Collection<FieldValue> fieldValues = new ArrayList<>();
        try
        {
            if (!(model instanceof ProductModel))
            {
                throw new FieldValueProviderException("Cannot evaluate originName of non-product item");
            }
            final ProductModel product = (ProductModel) model;
            List<String> rangeNameList;
            String productOriginName = getProductOriginName(product);

            rangeNameList = getRangeNameList(indexedProperty, productOriginName);
            final Collection<String> fieldNames = fieldNameProvider.getFieldNames(indexedProperty, null);

            for (final String fieldName : fieldNames)
            {
                if (rangeNameList.isEmpty())
                {
                    fieldValues.add(new FieldValue(fieldName, productOriginName));
                }
                else
                {
                    for (final String rangeName : rangeNameList)
                    {

```

```
            if (rangeNameList.isEmpty())
            {
                fieldValues.add(new FieldValue(fieldName, productOriginName));
            }
            else
            {
                for (final String rangeName : rangeNameList)
                {
                    fieldValues.add(new FieldValue(fieldName, rangeName == null ? productOriginName : rangeName));
                }
            }
        }
        catch (final Exception e)
        {
            LOG.error( message: "Failed to generate volume price", e);
            throw new FieldValueProviderException(
                "Cannot evaluate " + indexedProperty.getName() + " using " + this.getClass().getName(), e);
        }
        return fieldValues;
    }

    1 usage
    public String getProductOriginName(ProductModel product) {
        if (product == null || product.getProductOriginName() == null) {
            return null;
        }
        return product.getProductOriginName();
    }

    public FieldNameProvider getFieldNameProvider() {
        return fieldNameProvider;
    }
}
```

de.hybris.platform.solrfacetsearch.config.IndexedProperty

public String getName()

Platform Bootstrap (models.jar)

getName() on help.hybris.com

getName() on help.hybris.com

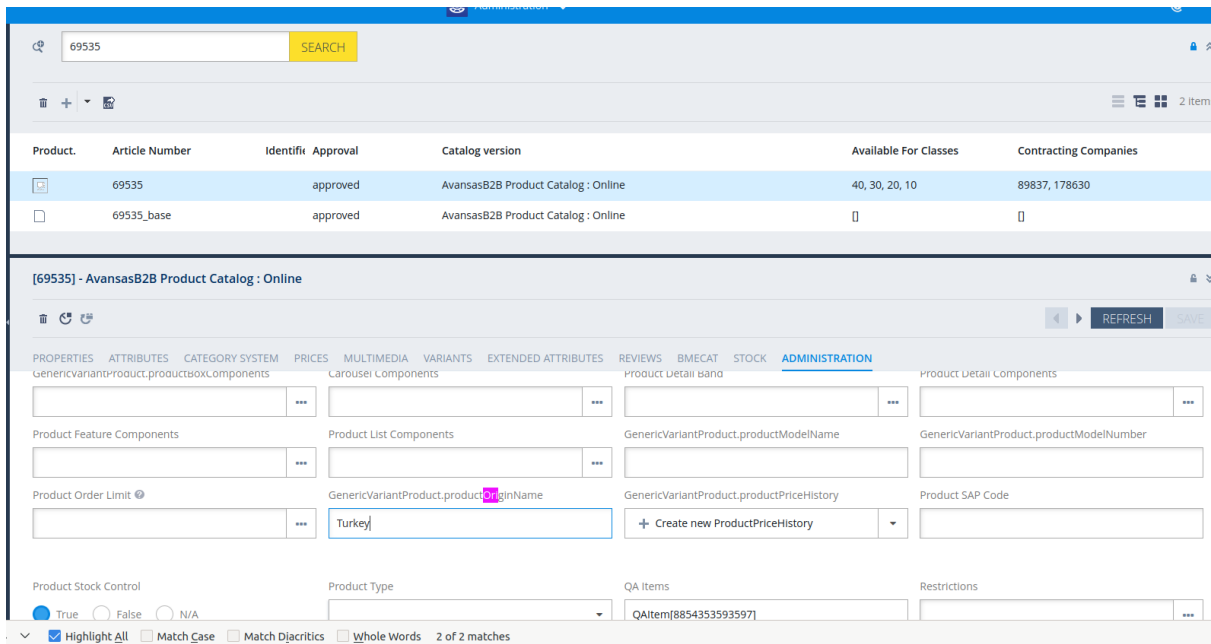
```
<bean id="productOriginNameValueProvider"
      class="com.avansas.solrfacetsearch.search.solrfacetsearch.provider.impl.ProductOriginNameValueProvider"
      parent="abstractPropertyFieldValueProvider">
    <property name="fieldNameProvider" ref="solrFieldNameProvider"/>
</bean>
```

Provider beans'i tanımlıyoruz. avansassolrfacetsearch-spring.xml'de

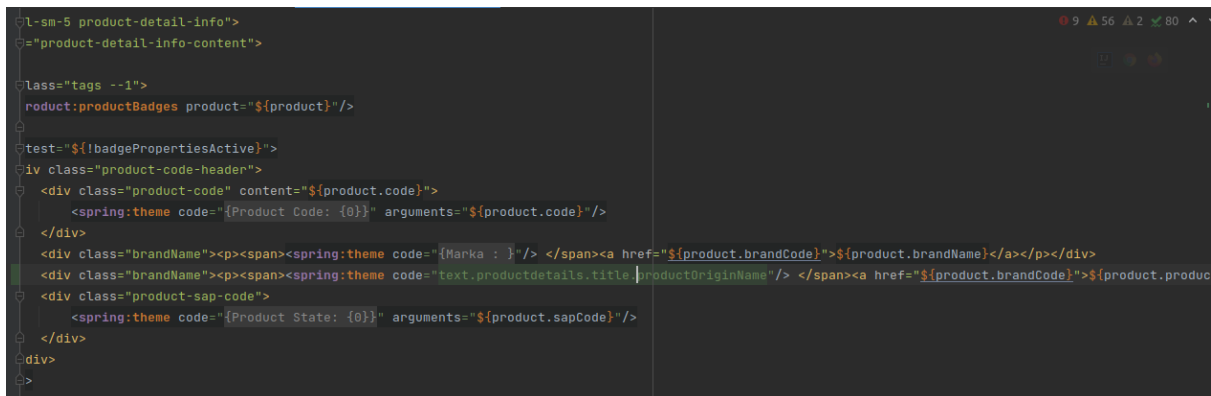
Ondan sonra sole.impex'de impex hazırlayıp basıyoruz :



Hybris backoffice'de 69535 kodlu olan urune productOriginName "Turkey" veriyoruz.



Ondan Sonra frontend'e gormemiz için ProductDetailsPanel'e productOriginName ekliyoruz.



başarılı gözükmetedir :



Ürün Kodu: 69535
Marka : Casio
Origin : Turkey

Casio MX-12B Masa Tipi Hesap Makinesi 12 Haneli

Listeye Ekle

Öne Çıkan Özellikler | Ürün Açıklamasına Git

- ✓ 12 haneli
- ✓ (Y) 29 mm x (G) 106,5 mm x (D) 147 mm
- ✓ Bağımsız hafıza
- ✓ Zam hesaplama
- ✓ Güneş enerjisi ve pil ile çalışma

95,79 TL KDV DAHİL

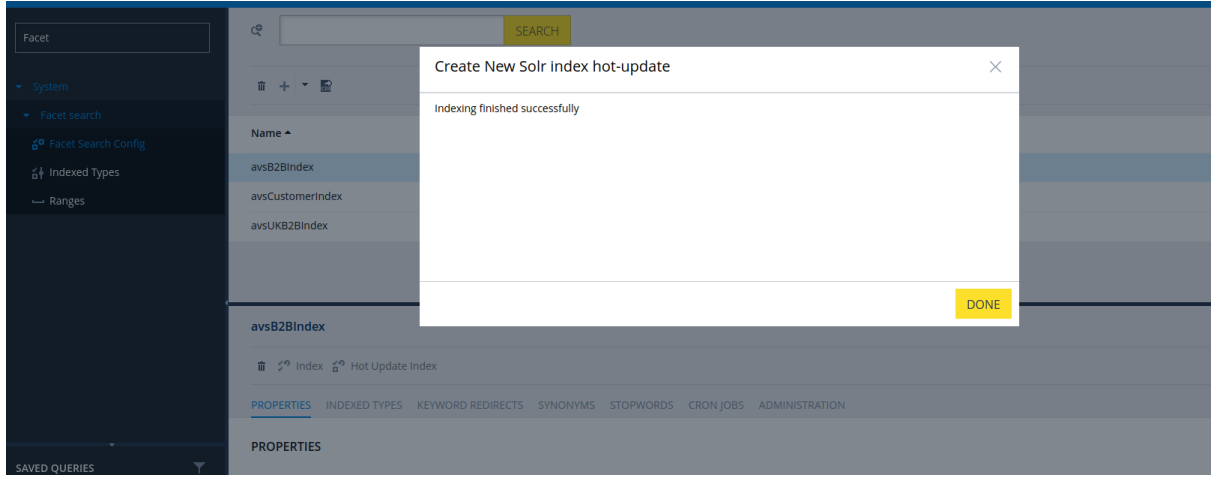
1 iş gününde teslimat detayları

1

Daha Fazla Hesap Makinesi

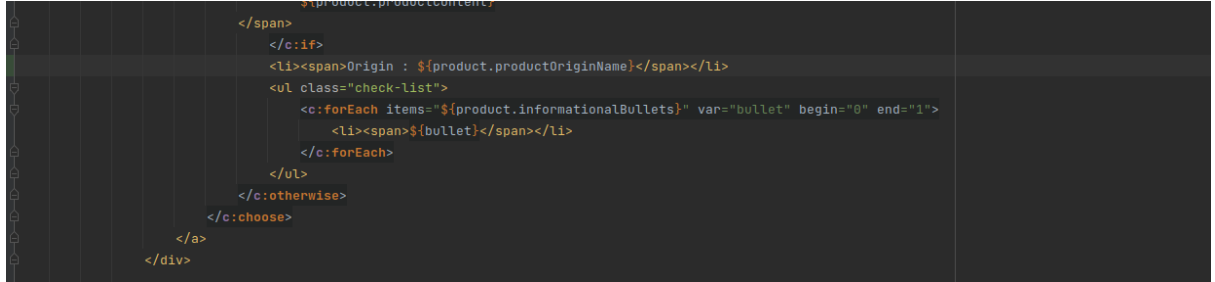
Daha Fazla Casio Hesap Makinesi

En son olarak ta product'a uyguladığımız değişiklikleri product listesinde görmemiz için Indexleme işlemi yapıyoruz .



Başarılı ile indexedik .

Frontende gelmesi için ProductListerGridItem.tag'a attribute ekliyoruz.



Toplam 1.165 sonuç



Ürün Kodu: 69535

**Casio MX-12B Masa Tipi
Hesap Makinesi 12 Haneli**

Origin : Turkey

- ✓ 12 haneli
- ✓ (Y) 29 mm x (G) 106,5 mm ...

95,79 TL KDV
DAHİL

—

1

+

SEPETE EKLE

Başarı ile listede görüldü .